BRAEMAR

APPLICATION NOTE

MANCHESTER PHASE ENCODING

APPLICATION NOTE

Spec. 0070

MANCHESTER PHASE ENCODING
as applied to
DIGITAL CASSETTE TAPE TRANSPORTS

Prepared by John C. Rooks

TABLE OF CONTENTS

1.0 INTRODUCTION

This application note is intended to familiarize the reader with the concepts necessary to interface a digital tape transport to a microprocessor system via the use of Manchester Phase Encoding.

This document is directed toward the person who has never worked with magnetic tape, and is totally unfamiliar with the associated recording techniques and terminology. As a result, there are several terms in this document which may not be familiar to the reader. When a new term is defined or explained, it will be entered in italics, and a statement clarifying the term will either precede or follow the term.

2.0 SATURATION RECORDING

The form of digital recording discussed will involve only saturation recording. In *saturation recording*, the magnetic field applied to the tape is strong enough to erase all previously recorded information. The user does not modify the intensity of the field. He can do only two things to it; turn it off and on, and alternate its polarity.

In Braemar transports, the presence or absence of the magnetizing field is determined by the "select read/write" line. The polarity of the field is determined by the "phase encoded data input" line. Both of these lines are, of course, digital, and respond to DC levels. A serial bit pattern applied to a phase encoded "data input" line during the "write" mode will be reproduced by the phase encoded "data output" line in the "read" mode.

3.0 ENCODING – AND THE REASON FOR IT

When dealing with signal lines intended for phase encoded data, the data output line reproduces the "data input" line. Since this is the case, an obvious way to implement a digital tape system is to simply feed and read a string of serial data and eliminate the encoding process. This would work only if the write and read speed were *absolutely* constant. Consider, for example, a series of 100 zeros followed by a one, all written at 1 ms intervals. If the read speed were 2% slow, the user, sampling at 1 ms intervals, would read 102 zeros rather than 100 before reading a one, and of course, the data content read would be incorrect.

As can be seen by the above example, it is also necessary to preserve the timing of the data. One method of accomplishing this is to *encode* the data as it is put on the tape in a fashion such that the content *and timing* of every bit is recognizable. An encoding scheme which reproduces the timing as well as the data is said to be *self-clocking*.

4.0 MANCHESTER PHASE ENCODING

Manchester Phase Encoding (also referred to in this document as phase encoding) is the accepted encoding system in the digital cassette industry. It is the system specified by the ANSI standard for full size cassettes (X3.48-1977). *Manchester Phase Encoding* is a self-clocking system. It requires at least one change in the level of the encoded data line for each data bit. This change (or *data transition*) associated with every bit is used to preserve the timing. The direction of the data transition is used to preserve the content of the data.

The "rules" for Manchester Phase Encoding follow:

- The data transition may be defined as being at beginning of the data cell.
- If the bit to be encoded is a one, the data transition is from a low to a high.
- If the bit to be encoded is a zero, the data transition is from a high to a low.

Consider an example where a transport is designed to operate at 2400 baud (bits per second). This means that data bits (or *data cells*) will have a width of 416 us, and that a new data bit will start every 416 us. A phase encoded byte of alternating zeros and ones would be observed on the phase encoded data lines as follows. (Incidentally, this is the *one* example where the encoding process does *not* alter the data wave form).

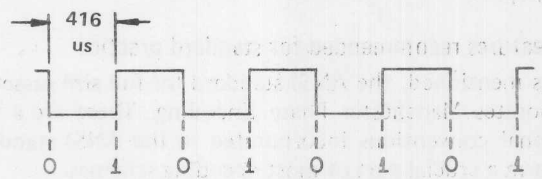

FIGURE 1.

Phase Encoded Byte 01010101

If the pattern continues and the next bits to follow are two zeros followed by two ones followed by four zeros, and if one sampled the phase encoded data lines every 416 us, he would observe level changes as indicated by the arrows in figure 2. As mentioned previously, these level changes are also known as data transition.

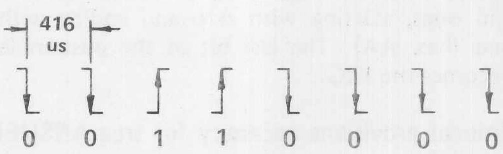

FIGURE 2.

In order to establish the data transitions shown in figure 2, it is sometimes necessary to incorporate intermediate level changes between the data transitions. These occur midway between data transitions and are known as mid bit transitions or *insignificant transitions* (rather than data transitions). The complete waveform (including mid bit transitions) for figure 2 is shown in figure 3.

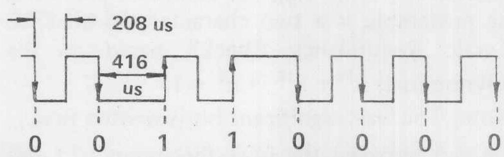

FIGURE 3.

Figure 3 is the Manchester Phase Encoded waveform for a pattern of 00110000, as it would be observed at the input and output of a machine which accepts and delivers Manchester Phase Encoded data.

5.0 ANSI/ECMA STANDARDS

The ANSI standard is a U.S. standard. The ECMA standard is a European standard. The differences between them are very subtle. The ECMA standard was written first with ANSI following by a few years. As a result, the ANSI standard clarifies some points which may be open to interpretation in the ECMA standard. In the real world, an ANSI compatible tape is an ECMA compatible tape.

The primary difference between the two standards is one of implication. ECMA implies that a compatible transport must have read while write capabilities. ANSI presents read while write as alternative and does not require or imply that read while write is necessary for compatibility.

5.1 Features recommended for standard practice

As mentioned, the ANSI standard for full size cassettes incorporates Manchester Phase Encoding. There are a few additional conventions incorporated in the ANSI standard that form a crucial part of most decoding schemes.

By formatting the *record* (or data block), using these conventions, encoding and decoding becomes much more convenient. (This will be discussed in detail later).

- *The Inter Record Gap (IRG)* is always written with a one (or high) polarity. This means that the space between data records is written with no transitions and is written with the data line high.
- Each record starts and ends with a *preamble* and *postamble*. The content of the preamble and postamble is eight bits of alternating zeros and ones, starting with zero and ending with one (hex AA). The last bit of the postamble becomes the IRG.

5.2 Additional provisions necessary for true ANSI/ECMA compatibility

The ANSI standard (like many specifications) is not made for casual reading. It is a very thorough document. Many users have asked the question "Can you tell me what else it requires so that I don't have to read it?" In full size cassette transports many of the ANSI compatibility requirements are the responsibility of the transport manufacturer. A list of user-influenced (or controlled) parameters also necessary to conform to ANSI compatibility follow:

- The IRG is a minimum of .7 inches and a maximum of 9.8 inches.
- Immediately following the data, and preceding the postamble is a two character, 16 bit CRC (Cyclic Redundancy Check), based on the polynominal) $x^{16} + x^{15} + x^2 + 1$.
  (Note: The least-significant bit is written first.)
- The gap between the BOT (beginning of tape) marker and the first record of data shall be 1.3 inches minimum and 9.8 inches maximum.
- The end of the last record may occur after the EOT (end of tape) marker; however, it must occur at least .7 inch before the end of recording media.

If these provisions and other additional provisions possibly made necessary by the particular ANSI compatible transport manufacturer are incorporated, the user will generate true ANSI compatible tapes that can be read by any ANSI compatible machine.

6.0 THREE QUARTER CELL TIMERS

The reader is now aware of the appearance of phase encoded data on the data lines, and the conventions used in the basic formatting of the data. The question of data decoding will now be discussed.

The objectives of a decoding scheme are to recognize the data transitions and ignore the mid bit or insignificant transitions. This is normally done with a *three quarter cell timer*. By starting a timer (either in hardware or software) with the first transition of the preamble, and masking out any further transitions which occur during the first three quarters of the data bit (data cell), mid bit transitions will be ignored and data transitions will be retrieved. The timer is re-started with each data transition. Figure 4 shows figure 3 repeated and the corresponding *window* pattern which results when a three quarter cell timer is employed. Only transitions during the shaded periods are considered as data; all others are masked out.

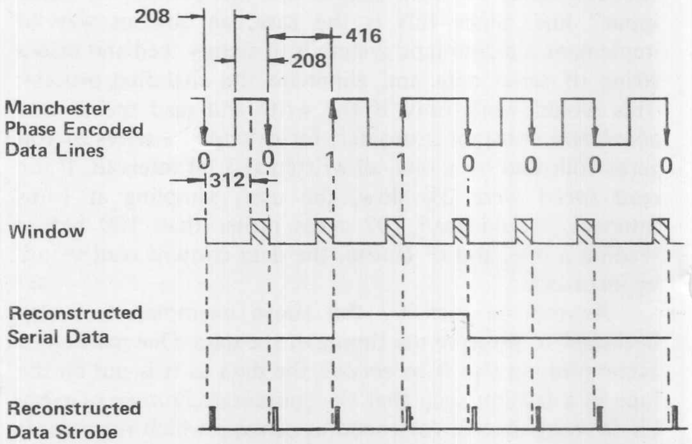

FIGURE 4
BYTE 00110000

By observing the direction of the transition of the encoded data at the beginning of each data bit, one can create "reconstructed serial data", and by observing the timing of the data transitions one can create "reconstructed data strobe" as shown in figure 4. These two "reconstructed" wave forms thus satisfy the objectives of phase encoding in that both the content and timing of the data have been preserved.

6.1 Fixed Three Quarter Cell Timer

The simplest form of a three quarter cell timer is a fixed three quarter cell timer. This system opens a "window" for a fixed number of microseconds after the data transition. This scheme is routinely employed with digital counters, one shots, or software counters. It is vulnerable to timing errors from many different sources. If the combination of these errors shifts a mid bit into the shaded area (window) of figure 4, an error will result.

Specifically, this scheme is susceptible to the following sources of timing error.

- Accuracy of time base used in write.
- Accuracy of time base used in read.
- Long term changes in machine speed (i.e., speed differences observed between write and read).
- Speed differences between machines. (Of concern only if tape is written and read on different machines).
- Short term, or bit to bit speed variations (*jitter*)
- Cycle time of microprocessor.

In spite of the numerous sources of error, users are employing a fixed three quarter cell timer and observing error-free operation.

6.2 Variable Three Quarter Cell Timer (set by preamble)

If the user desires a greater margin of safety (with respect to timing errors) he may employ a variable three quarter cell timer. This approach minimizes the impact of the first four sources of error listed in section 6.1.

It was mentioned previously that the preamble was "convenient" for some decoding schemes. This "convenience" will now be utilized. Since the preamble is 8 bits of alternating zeros and ones, *it contains no mid bit transitions*. This means that the spacing of the first eight transitions of every record are representative of the bit spacing that will be observed as data flows off the tape. If the user measures the spacing of the transitions of the preamble and uses the value obtained to determine the opening point of the window, he is using a variable three quarter cell timer.

This scheme compensates for time base variations, long term speed changes and machine to machine speed variations.

6.3 Variable Three Quarter Cell Timer (set by previous bits)

An additional degree of sophistication and reliability can be obtained by proceeding yet one step further. Once the value of time between preamble transitions has been obtained, it may be used to anticipate the position of the succeeding data transitions. If the spacing between the data transitions is then measured and used in a continuous update mode, the user has one of the most reliable systems available. Such a scheme literally anticipates the positon of the next data transitions by measuring the spacing of the previous transitions.

6.4 Variable Three Quarter Cell Timer (set by the average of previous bits)

The final three quarter cell to be discussed will be an elaboration of the last system presented above. To describe this system we shall consider the case where a data transition appears to be shifted from its intended location on the tape. Such a shift may occur if a slight flow existed on the tape, or if a noise pulse causes a premature indication of a data transition.

Such a shifted transition would result in an apparent short cell followed by an apparent long cell. If the timer were set by the previous bit (as in the last method above) it would anticipate a short bit following a short bit, whereas in the case of a shifted transition, it *should* actually anticipate a long bit following a short bit. The way out of this dilemma is to average the time over the last few transitions and then use the average value of cell time to set the timer. In hardware, this may be accomplished as easily as adding an RC network. In software, it will require some additional operations.

Whichever method is chosen is up to the user. All of the methods above are used, and all are satisfactory in particular applications. All of the three quarter cell timing techniques presented can be implemented in either hardware or software. Previous examples have discussed a data rate of 2400 baud. This rate is slow enough to allow most microprocessors sufficient time to manipulate the data between data transitions. Most users at this slow rate choose a software timing technique (assuming adequate memory is available). It is normally the most economical system.

6.5 Should a Three Quarter Cell Timer be a Two-Thirds Cell Timer?

To most observers, the opening of the window, in figure 4, at the three quarters point is a very logical thing to do. If the time base (i.e. window opening point) shifts, the point of greatest safety margin is midway between the insignificant transition and the data transition and a three quarter cell timer therefore makes good sense. Again, the objective is to open the window at a point where the probability is maximized of masking out an insignificant transition and detecting a data transition.

There is a fallacy in the above argument. The above assumption is true only if the time base (or window opening point) changes. In reality, time bases are very stable compared to tape speed changes. Since the tape speed is really what changes, one must evaluate this situation from a different perspective.

Speed variations which cause a cell width variation of $\pm\delta$ will result in the waveform of figure 4A, where τ is the cell width.

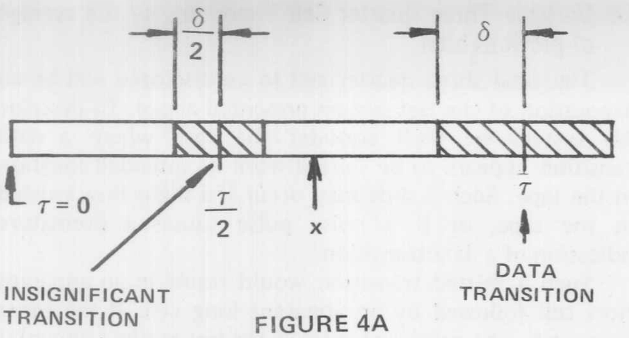

FIGURE 4A

Note that the effect of speed variation on the mid bit transition is only half that of the data transition, (i.e., $\delta/2$). The optimum spot to open the window (X) is defined as the point where the speed variation, and its associated period variation (δ) can be the greatest value (δm) without either opening the window – before the insginificant transition – or after the data transition.

By inspection:

$$X = \frac{\tau}{2} + \frac{\delta m}{2} \qquad X = \tau - \delta m$$

Solving for δm $\qquad \delta m = \tau/3$

Solving for X: $\qquad X = \frac{2}{3}\tau$

The rather surprising conclusion from this derivation is that a three quarter cell timer should really be a "two thirds" cell timer. And yes, when applied in situations where the primary variable is machine speed, this is true.

For optimum results, the window should open approximately two thirds of the way through the data cell rather than at the three quarter cell point. This is contrary to accepted practice in many parts of the industry.

7.0 ENCODING CONSIDERATIONS

Encoding (putting the data on the tape) is normally a simple operation. If done in hardware it may be accomplished with two type D flip flops and an exclusive NOR gate as shown in figure 5.

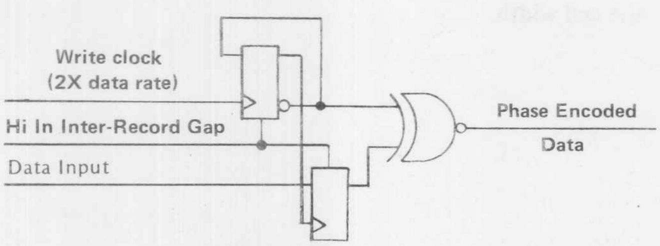

FIGURE 5.

Manchester Phase Encoding Circuit

If the encoding is done in software, the programmer may find it useful to observe a phase encoded waveform "shifted" by 180°.

In figure 3 and 4 we presented the byte 00110000 and illustrated it as indicated in figure 6 (repeat of figure 3).

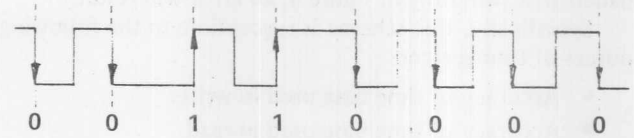

FIGURE 6.

Phase Encoded Byte 00110000

Previously we have defined the data transition as occurring at the beginning of the data cell. This was an arbitrary choice for clarification of phase encoding. The data transition could just as easily have been considered in the center of the data cell. If figure 6 is kept intact and the observer shifts his perspective of the cell by one half cell, a rather interesting pattern developes (see dotted lines in figure 7). Keep in mind the content and timing of figure 6 and figure 7 are identical; only the observer's perspective (as shown by the dotted lines) has changed.

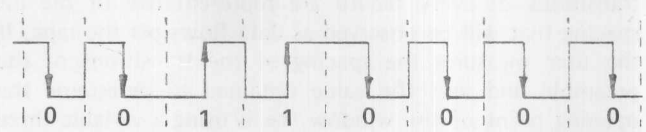

FIGURE 7.

Phase Encoded Byte 00110000

Using inspection, a Manchester Phase Encoded waveform may also be defined as follows:

- The level of the second half of the cell is always the value of the data prior to encoding.
- The first half of the phase encoded cell is always the complement to the second half of the phase encoded cell.
- Therefore, the level of the data in the first half of the cell is the inverse of the data prior to the encoding.

By considering a phase encoded format as described above, the programmer may find his encoding operations simplified.

8.0 ENCODING AND DECODING AS APPLIED TO BRAEMAR TRANSPORTS

8.1 CM600 Mini Dek

The CM600 accepts and delivers phase encoded data. The phase encoded waveform fed to its "data input" line in the "write" mode is reproduced at its "data output" in the read mode. Any of the decoding methods previously discussed may be employed with this unit. It is a very versatile instrument. It will accept data rates much below the 2400 baud specified. If the user employs a lower rate he should not go below the limits of the coupling capacitors in the read amplifiers. The voltage across these capacitors must not change. The upper limit of the machine is dependent upon the tape density and noise rejection filters. The nominal minimum design pulse width is 208 us and must not be less than approximately 185 us.

Since the CM600 reproduces its input at its output and is limited only by minimum and maximum transition spacings and speed variation, there are several other encoding systems which will operate satisfactorily with the CM600. Several customers are employing alternate schemes successfully. Please consult the factory if additional information is desired.

In the event the user is tempted to feed asynchronous bytes to the CM600 (such as derived from a UART) without encoding, he should be cautioned that this scheme "almost works". And if by chance the particular machine he experimented with did work using this system, chances are very good that the next machine he obtains will not perform similarly. Save the frustration. Encoding formats were created for a reason. They *are* necessary. Please use them.

8.2 CD200 Tachometer Controlled Digital Cassette Transports

This unit is a full size cassette transport which can employ an ANSI compatible format. It has a digital head and no additional electronics other than motion control. The nominal data rate at 10 ips is 8K baud. The user must provide read/write amplifiers, as well as any encoding and decoding systems. All of the decoding systems presented can be made to operate with this unit; however, the fixed three quarter cell timer is not recommended.

The data rate is high enough (8K at 10 ips, 16K at 20 ips) that most users do not attempt software decoding, and rely on hardware schemes. There is no lower limit on the data rate of this unit. As with the CM600, encode/decode schemes are necessary with the CD200.

8.3 CS410A (with read/write amplifiers)

This is a CS200 with read/write amplifiers added. Its capabilities and limitations are similar to those listed with the CD200.

8.4 CS400A (with read/write and encode/decode electronics)

This is a CD200 with all the read/write and encode/decode electronics provided. The unit accepts and delivers serial digital data. No additional data conditioning or encoding is necessary. It is an extremely reliable device and employs the most sophisticated of the decoding schemes described in the schemes described in the text of this document.

8.5 CS400B (with read-while-write electronics)

This unit operates in identical fashion to the CS400A, with the additional feature that it operates in a read-while-write mode. This feature provides instant verification of data as it is written on the tape. The data inputs and outputs are the same as on the CS400A. As with the CS400A, all encoding and decoding is done within the transport and the user communicates with serial digital data.

8.6 LTD – Long Term Datalogger

This unit is extremely low speed and low power. It can be set to run for days at a time on one cassette. It is supplied with a variety of heads and can employ a variety of writing schemes, depending on the application.

9.0 SUMMARY

This document has summarized the details of Manchester Phase Encoding as it applies to Braemar digital cassette tape transports. By employing the methods described, the user will have reliable error-free operation. If additional information is required, please contact the factory.

BRAEMAR
COMPUTER
DEVICES, INC.

11950 TWELFTH AVENUE SOUTH • BURNSVILLE, MINNESOTA 55337 • AREA (612) 890-5135

3/78